



HAL
open science

Bond Graphs - A graphical language for the analysis of multiphysical systems

Pierre Haessig

► **To cite this version:**

Pierre Haessig. Bond Graphs - A graphical language for the analysis of multiphysical systems. Engineering school. France. 2022. hal-03602684

HAL Id: hal-03602684

<https://cel.hal.science/hal-03602684>

Submitted on 28 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Bond Graphs

A graphical language for the analysis of multiphysical systems

→

Pierre Haessig, CentraleSupélec (campus of Rennes)

SG6: Dec. 2021 – Jan. 2022

Slide deck 1: the language

Course outline

- **Bond graph objectives**
- **The bond graph language**
 - Bonds and power variables: the physical analogy
 - Elements
- Practice: reading & creating bond graphs
- Causality and derivation of mathematical models

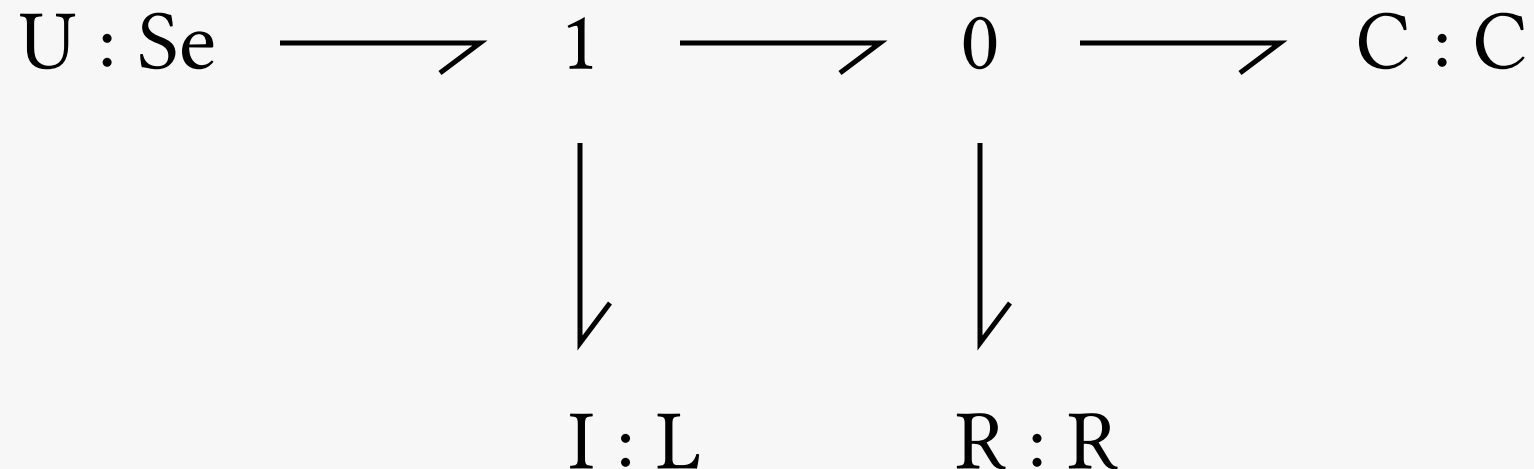
Bond graph objectives

- a simple unified graphical language for many physical domains
- acausal models
 - to preserve the physical structure of the real physical system
 - which highlight energy exchange
- but with the (optional) superposition of a *computational causality* information

Bond graph model structure is *hybrid*

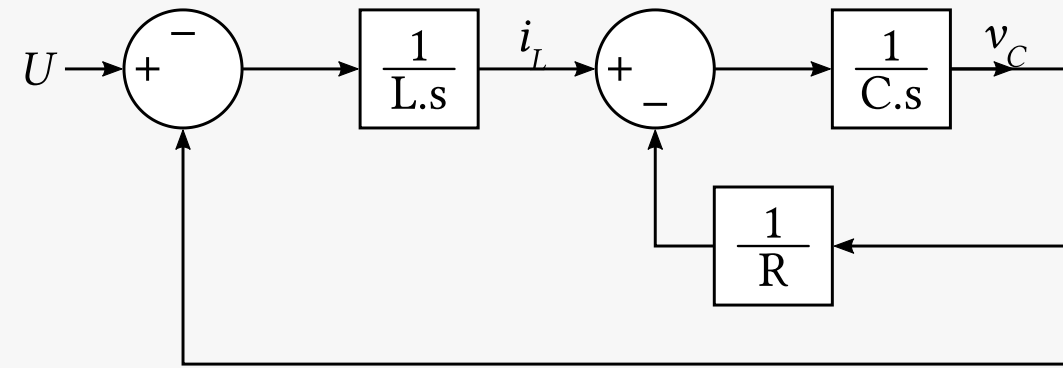
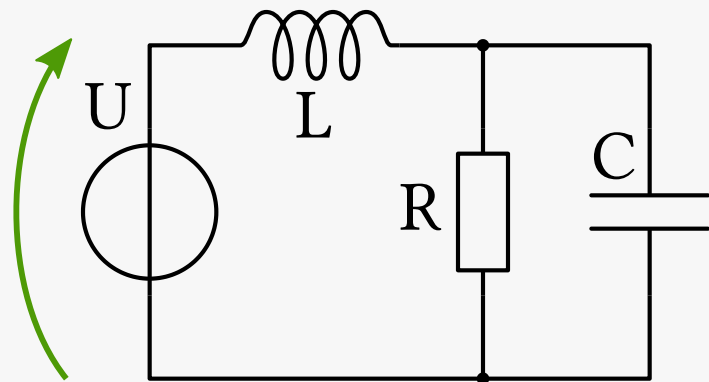
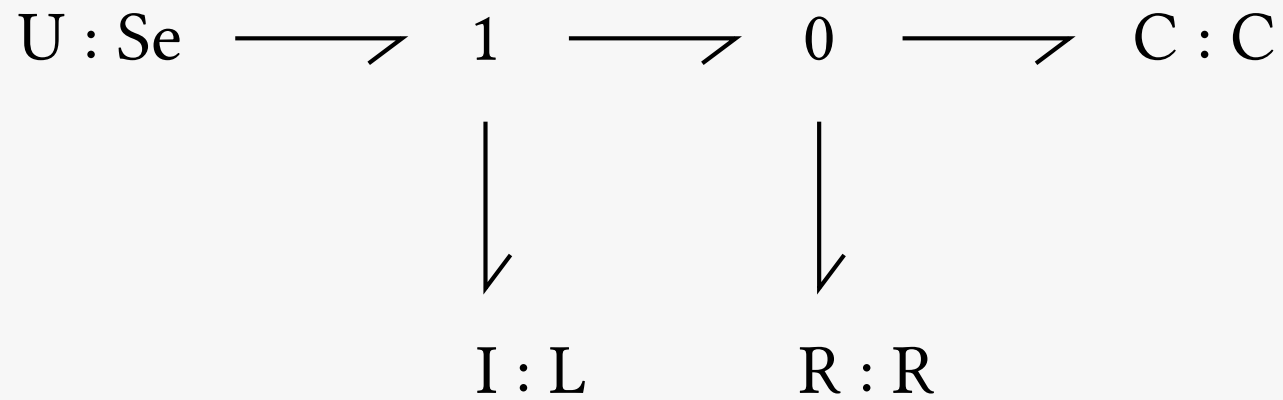
Two extremal structures of model:

1. Block diagrams, with very clear computational structure
 - but lost physical structure
2. Physical network-type diagrams (electrical, mechanical)
 - but no computational information



Model comparisons

BG, Circuit (acausal), Block diagram (causal)

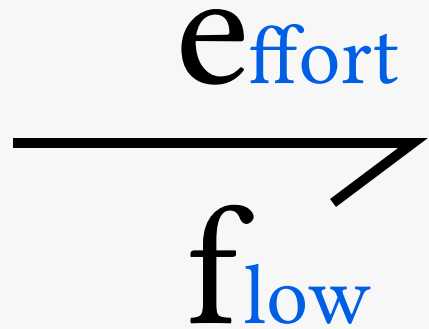


The bond graph language

- Bonds & power variables
- Elements

Bonds

Bonds model the physical interaction of **two** elements which **exchange energy**.



The interaction happens through two generalized physical variables: *effort* & *flow*, collectively named the “power variables”.

Physical analogy in bond graphs

Each physical domain has a specific choice for the generalized *effort* and *flow* variables of each bond:

	Effort e	Flow f
Translational mechanics	Force F (N)	Velocity v (m/s)
Rotational mechanics	Torque Γ (N.m)	Angular velocity ω (rad/s)
Electricity	Voltage u (V)	Current i (A)
Thermal transfers	Temperature T (K)	Entropy flow rate \dot{S} (J/K/s)
Hydraulic	Pressure P (N/m ²)	Volume flow rate Q_v (m ³ /s)

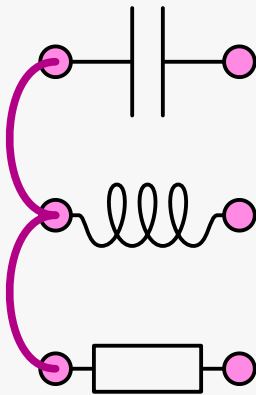
Property: the two variables of each bond are chosen such that:

$$Effort \times Flow = Power \text{ (Watt)}$$

Reminder: Modelica's physical analogy

Modelica's analogy is based on the port's *connection* behavior

	Potential	Flow
Translat. mech.	Position s (m)	Force f (N)
Rotational mech.	Angular position φ (rad)	Torque Γ (N.m)
Electricity	Voltage u (V)	Current i (A)
Thermal transfers	Temperature T (K)	Heat flow Q (J/s=W)
Hydraulic	Pressure P (N/m ²)	Mass flow rate Q_m (kg/s)



$$\begin{aligned} V_1 &= V_2 = V_3 \\ i_1 + i_2 + i_3 &= 0 \end{aligned}$$

⚠ Differences with bond graph:

- Force and Torque are **switched**: BG's Effort \rightarrow Modelica's Flow
- Different vocabulary in **bold**: in particular position vs speed

Comparison with Modelica's analogy

BG & Modelica:

- both introduce an analogy between variables across different physical domains
- but using a different **classification** & **vocabulary**, because each is built on different foundations

BG's analogy: group variables as “*effort*” or “*flow*”

- by *preserving common physical sense* (ex.: voltage \leftrightarrow force, current \leftrightarrow speed)
- with constraint $e \times f = \text{Power}$

Modelica's analogy: group variables as “*potential*” or “*flow*”

- by *preserving the connection topology* of graphical diagrams (ex.: voltage \leftrightarrow position because both are equal at interconnection of ports)
- (with no constraint on the product $e \times f$)

Relation between BG's and Modelica's analogies

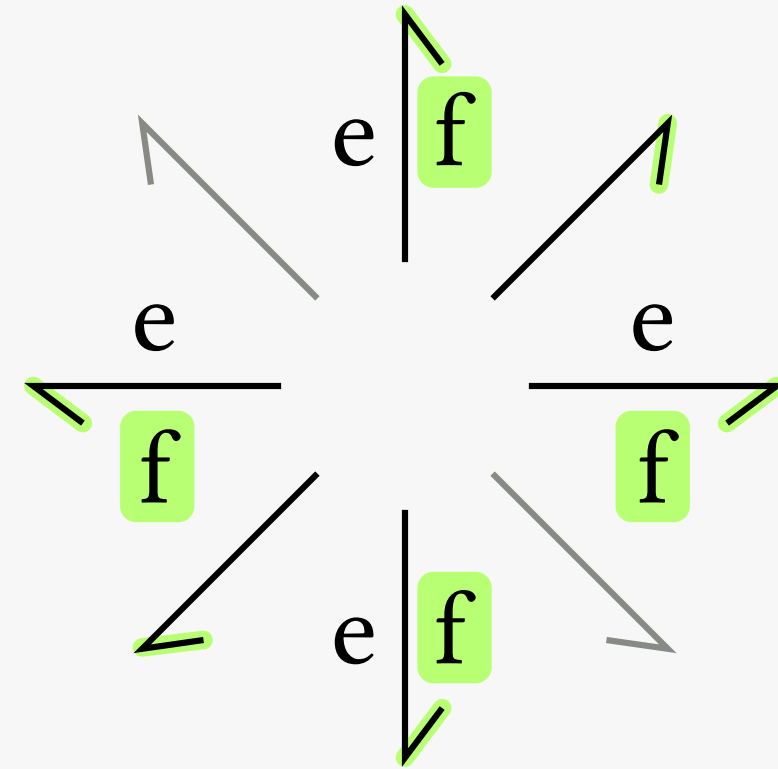
Domain	Mechanics	Others (e.g. electricity)
BG effort	Mod flow	Mod potential
BG flow	Mod der(potential)	Mod flow (*)

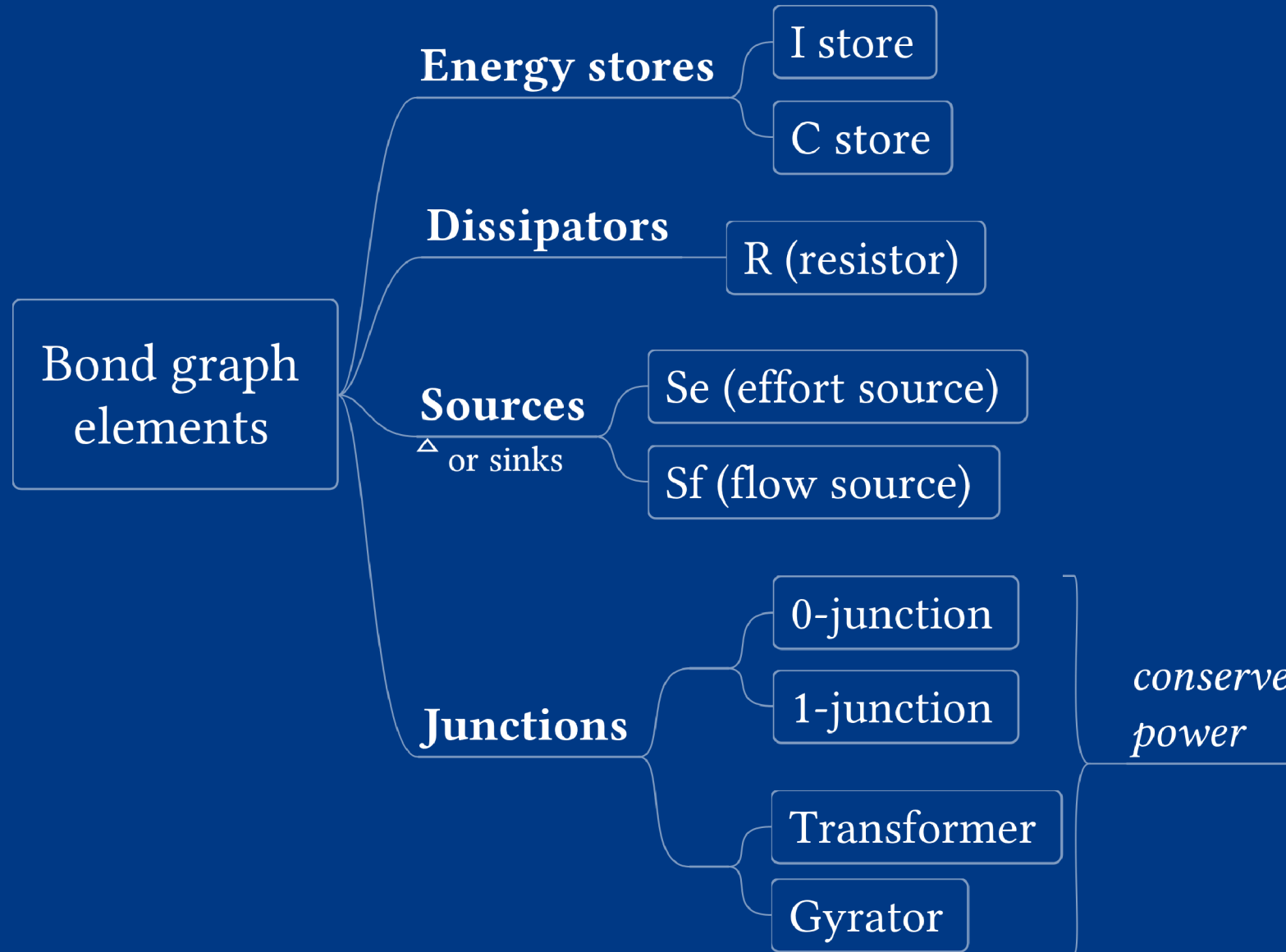
Observations:

- for all domains except mechanics, we have:
 - BG effort = Modelica potential
 - BG flow = Modelica flow (*)
- but for mechanics:
 - it's reversed
 - and we have an extra derivative: speed = der(position)
- (*) thermal domain is an exception : heat flow (J/s) vs entropy flow (J/K/s)

Bonds

- Exchange of energy between elements
- Half arrow = direction of positive power flow
- Drawing conventions:
 - e above/left, f below/right
 - half arrow on the side of the flow (i.e. below/right)





Junctions

Unlike in network-type diagrams, the connection of elements is not achieved with the topology of links (e.g. loops of wires), but using explicitly one of the two junction elements:

- “0” junction: common effort
- “1” junction: common flow

Also in the junction category: transformers and gyrators

Common property: **instantaneous power is conserved**

0 junction

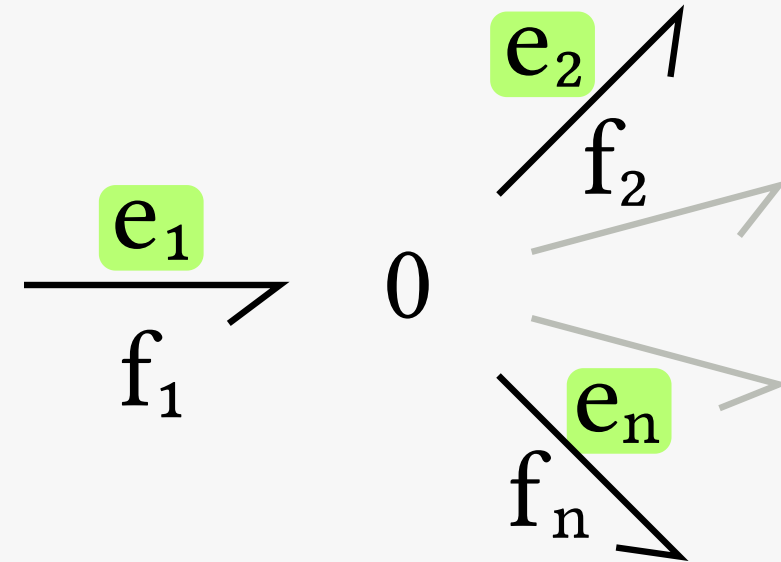
“Common effort” junction:

$$e_1 = e_2 = \dots = e_n$$

Flows are distributed (incoming sum = outgoing sum), according to the *orientation* of the bonds.

On the example:

$$f_1 = f_2 + \dots + f_n$$



1 junction

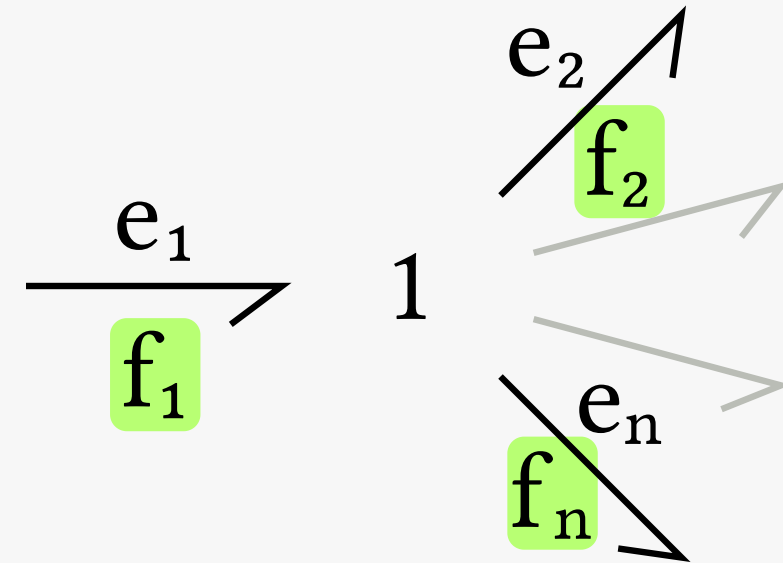
“Common flow” junction:

$$f_1 = f_2 = \dots = f_n$$

Efforts are distributed (incoming sum = outgoing sum), according to the *orientation* of the bonds.

On the example:

$$e_1 = e_2 + \dots + e_n$$

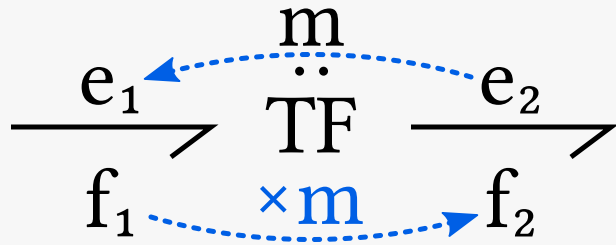


Two-port junctions: transformers & gyrators

Transmit power with a scaling of efforts & flows:

- in the same domain (unitless scaling)
- between two domains (scaling with a physical unit)

Transformer (TF)



$$f_2 = m \cdot f_1$$

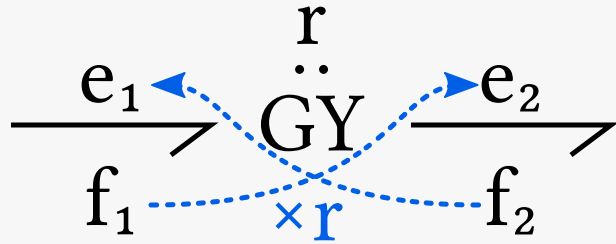
$$m \cdot e_2 = e_1$$

m is the transformer ratio

Examples:

- Mechanical gear pair: $\omega_2 = (r_2/r_1) \cdot \omega_1$
- Cable – Pulley: $v = r \cdot \omega$
- Electrical **transformer**: $v_1 = m \cdot v_2$ (⚠ inverted definition of the transformer ratio)

Gyrator (GY)



$$e_2 = r \cdot f_1$$

$$e_1 = r \cdot f_2$$

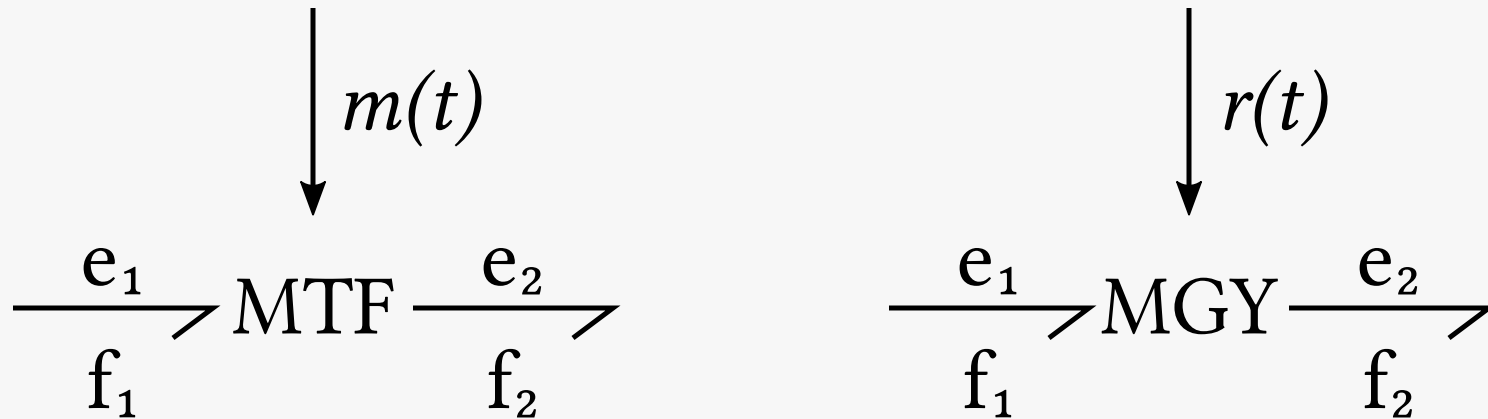
r is the gyrator ratio

Example:

- EMF of a DC motor: $e = K \cdot \Omega$ and $C = K \cdot i$

Modulation of Transformers (Gyrators)

The transformer (gyrator) can be *modulated by a signal* (using a *signal arrow* \rightarrow):



Examples: crank-slider mechanism, averaged DC-DC converter.

One-port elements

There are 3 basic energy consuming/storing devices:

- Dissipator: **R**esistor
- Energy stores:
 - **C** store, also called **C**ompliance or **C**apacitor
 - **I** store, also called **I**ertia

In addition, there are two sources: **S_e** (effort) and **S_f** (flow source)

Remark: in electricity, *one-port* element = device with *two* electrical pins

Resistor (R)

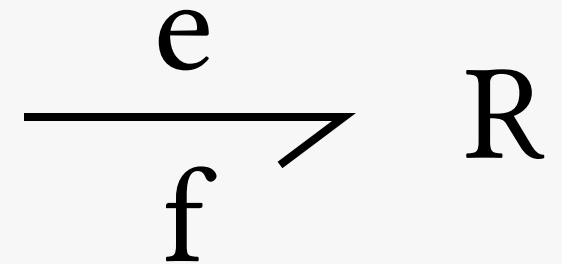
Relation (linear case):

$$e = R.f$$

Property: power is irreversibly dissipated (as heat)

Examples:

- Electrical resistor: $u = R.i$
- Mechanical damper: $f = d.v$



C energy store (also called Compliance or Capacitor)

Using the “generalized displacement” q (an “energy” variable):

$$q = \int f \cdot dt$$

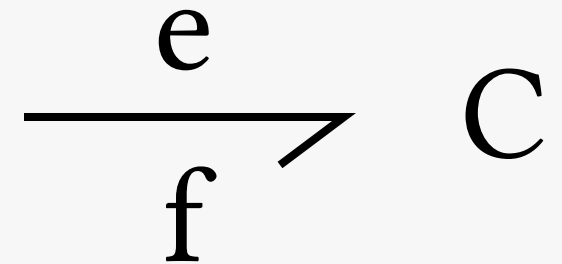
C store relation:

$$q = \Phi_C(e)$$

Linear C store:

$$q = C \cdot e$$

→ consequence: $f = C \cdot de/dt$



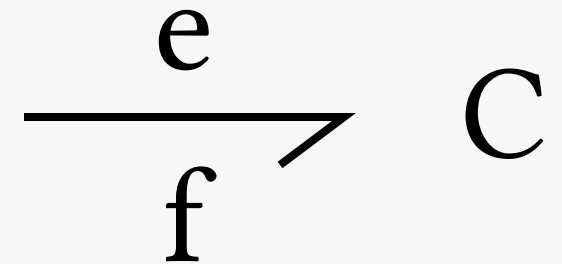
C store examples (linear)

Mechanics:

- displacement = (kinematic) displacement $x = \int v \cdot dt$
- relation: $x = (1/k) \cdot f$
- C store = **spring**

Electricity:

- displacement = charge $q = \int i \cdot dt$ (Coulomb)
- relation $q = C \cdot u$
- C store = (electrical) **capacitor**



I energy store (also called Inertia)

Using the “generalized momentum” p (an “energy” variable):

$$p = \int e \cdot dt$$

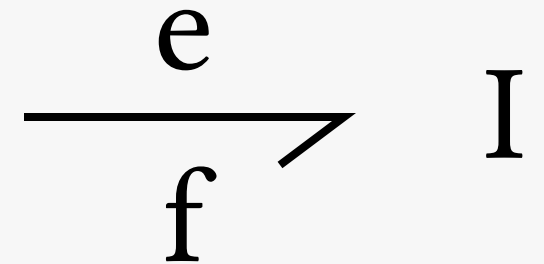
I store relation:

$$p = \Phi_I(f)$$

Linear I store:

$$p = I \cdot f$$

→ consequence: $e = I \cdot df/dt$



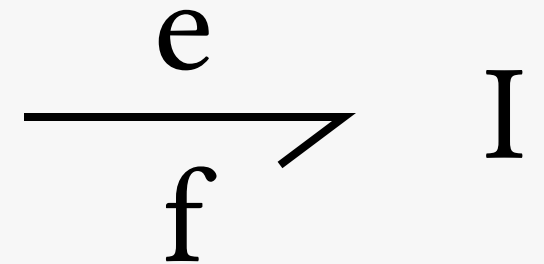
I store examples (linear)

Mechanics:

- momentum = mechanical momentum $p = \int f \cdot dt$
- relation: $p = m \cdot v$, that is $f = m \cdot dv/dt$ (inertial force)
- I store = mechanical **inertia**

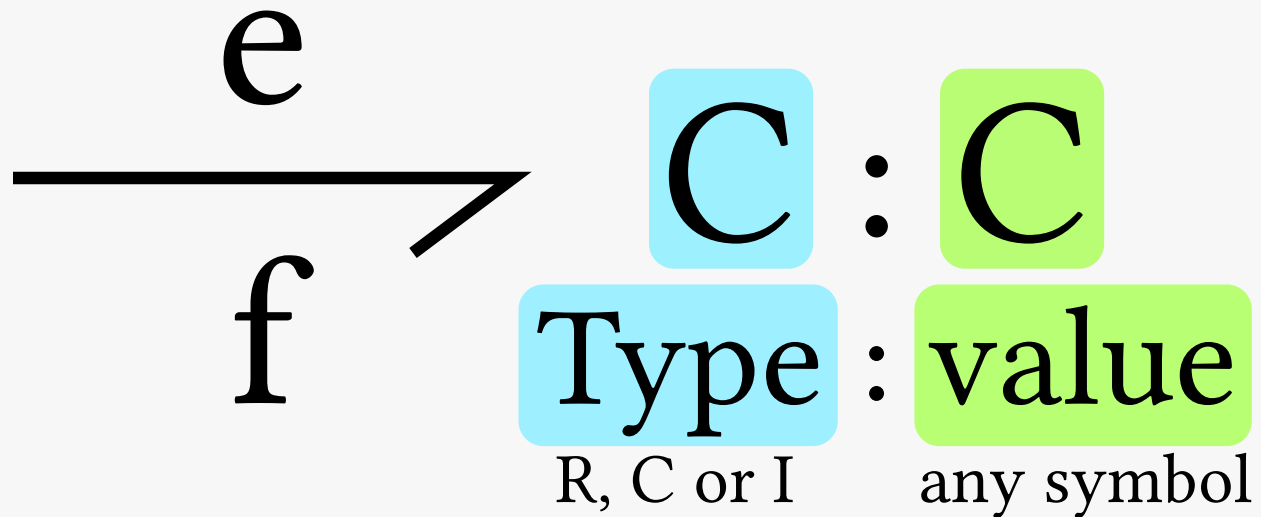
Electricity:

- momentum = magnetic flux linkage $\lambda = \int u \cdot dt$ (V.s = Wb)
- relation: $\lambda = L \cdot i$, that is $v = L \cdot di/dt$
- I store = **inductor**



Parametrization of R, C, I elements

The value of a linear R/C/I element is appended with the notation “: x ”.



Same notation is used for the value of a source (next slide).

Sources

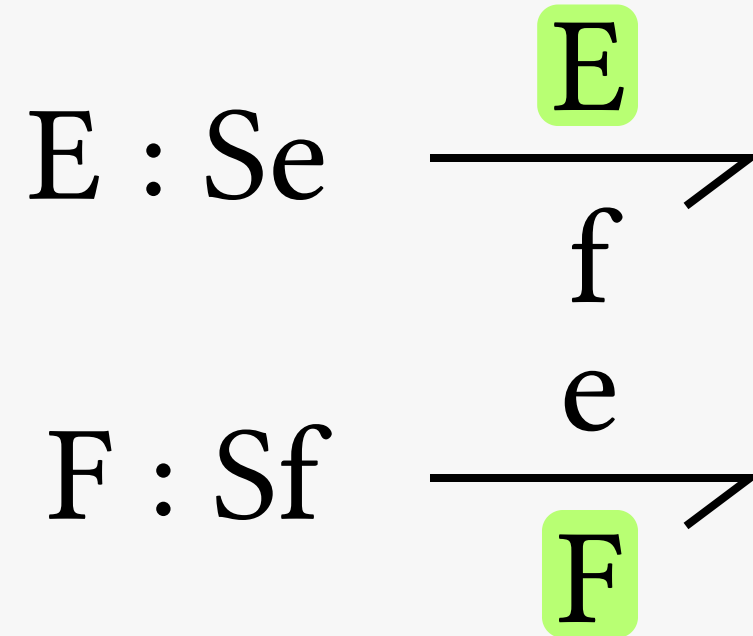
Sources either impose the effort (**Se**) or the flow (**Sf**).

Se examples:

- Electricity: voltage source
- Mechanics: imposed torque or force

Sf examples:

- Electricity: current source
- Mechanics: imposed speed



Bond Graphs practice

A graphical language for the analysis of multiphysical systems

→

Pierre Haessig, CentraleSupélec (campus of Rennes)

SG6: Dec. 2021 – Jan. 2022

Slide deck 2: practice

Course outline

- Bond graph objectives
- The bond graph language
 - Bonds and power variables: the physical analogy
 - Elements
- **Practice: reading & creating bond graphs**
- Causality and derivation of mathematical models

Practice

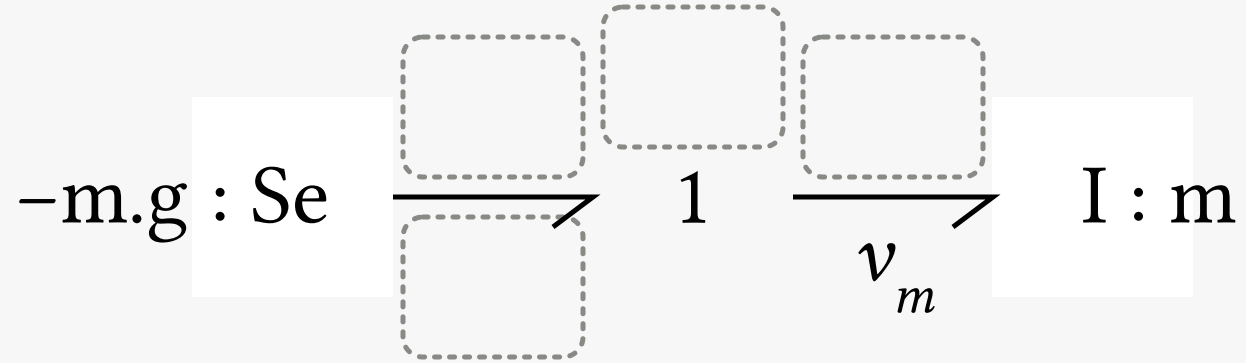
Objectives: being able to

1. Read BGs

- recognize classical structure

2. Create BGs from network diagrams (electrical, mechanical)

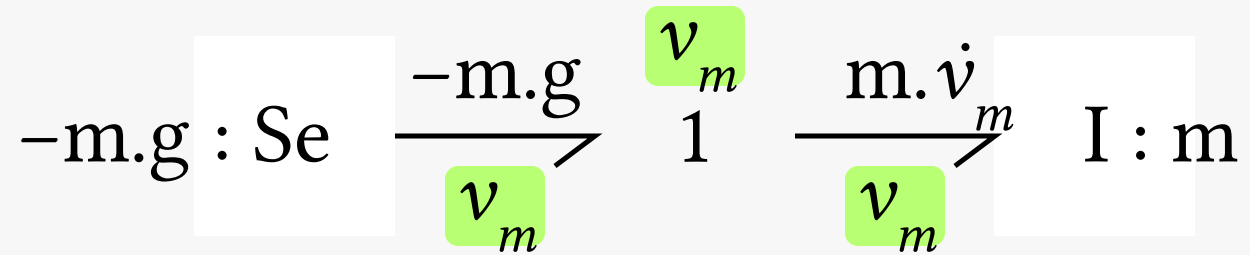
M1: 1 junctions represent “velocity points”



Questions:

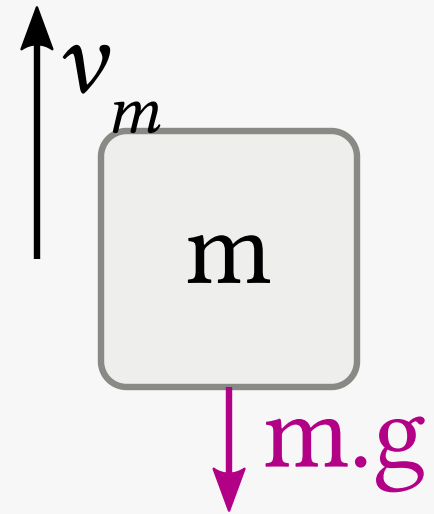
- Fill the blanks on the BG
- Represent the physical system
- Derive the overall mechanical equation of the system

M1: 1 junctions represent “velocity points” (S)

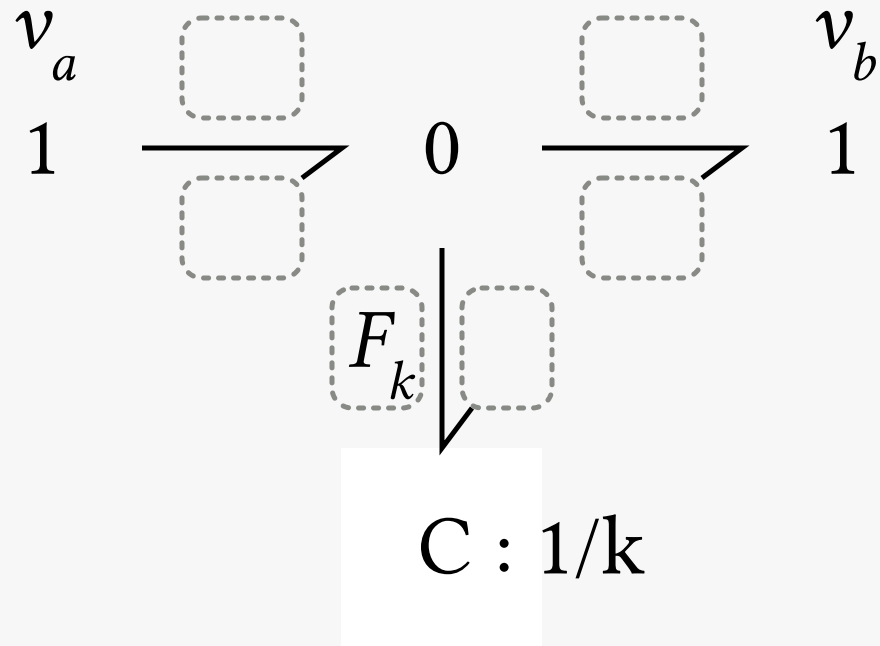


This BG represents the **free fall** of a mass m with equation:

$$m \cdot \frac{dv_m}{dt} = -m \cdot g$$



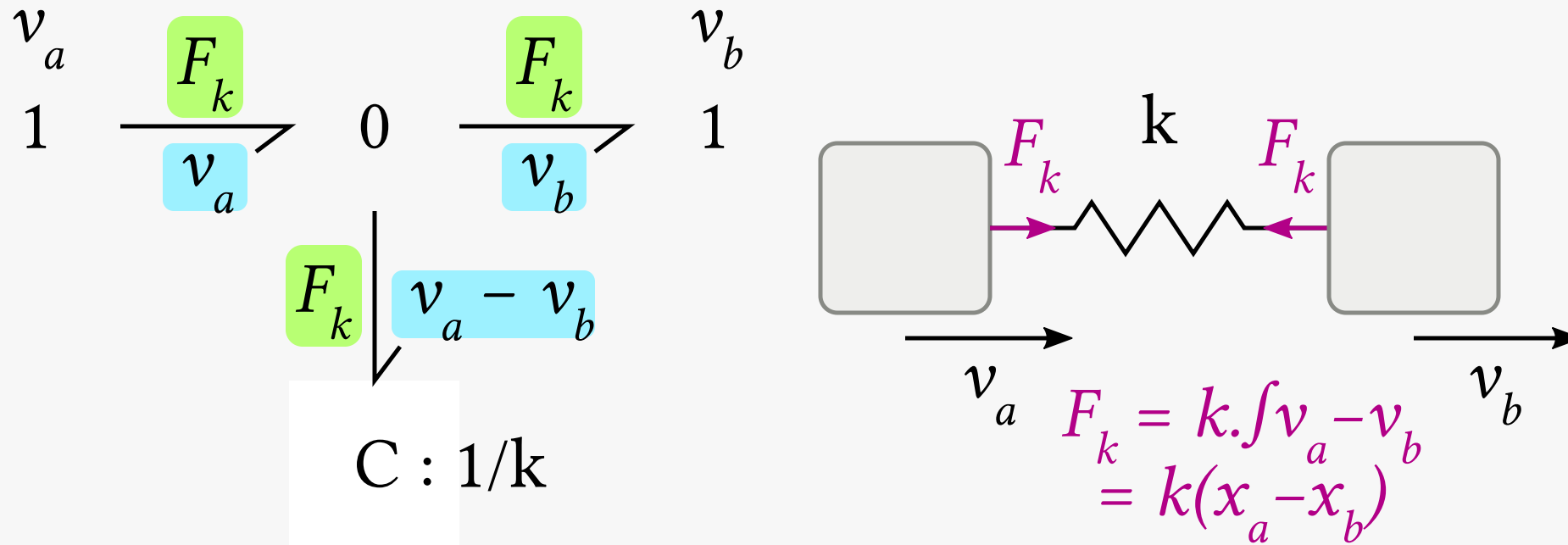
M2: Springs and dampers typically on 0 junctions



Questions:

- Fill the blanks on the BG
- Represent the physical system
- Give the expression of F_k

M2: Springs and dampers typically on 0 junctions (Sol.)



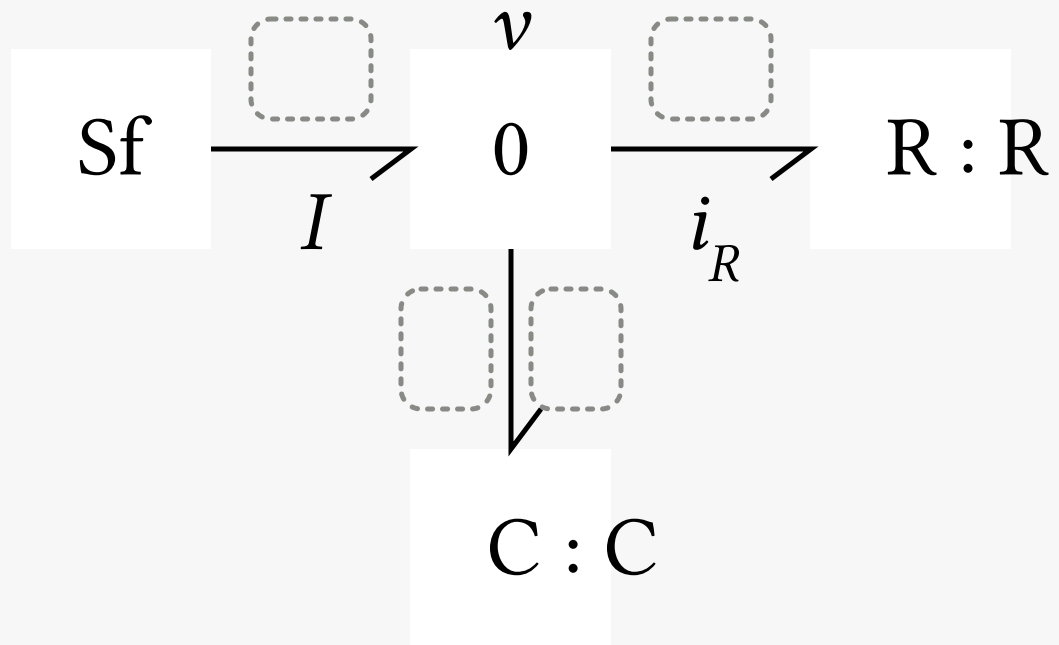
This BG represents a spring of stiffness k between two points A and B.

$$F_k = k \int (v_a - v_b) \cdot dt = k(x_a - x_b)$$

Mechanical BGs: main structures

- velocity points represented by 1 junctions
 - an Inertia can attach directly to its corresponding 1 junction
- components which reacts on a velocity/position differences (spring and dampers) are typically connected to a *0 junction placed between two 1 junctions*

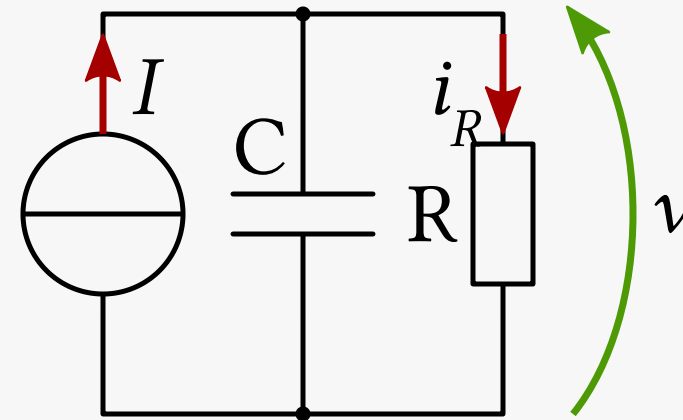
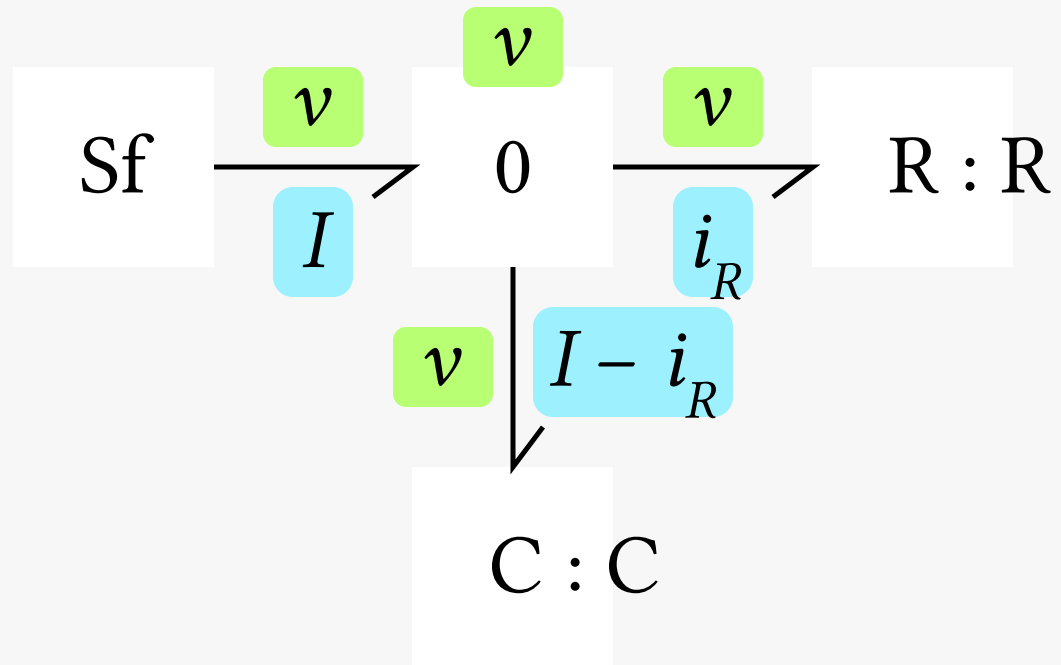
E1: 0 junctions represent voltage nodes



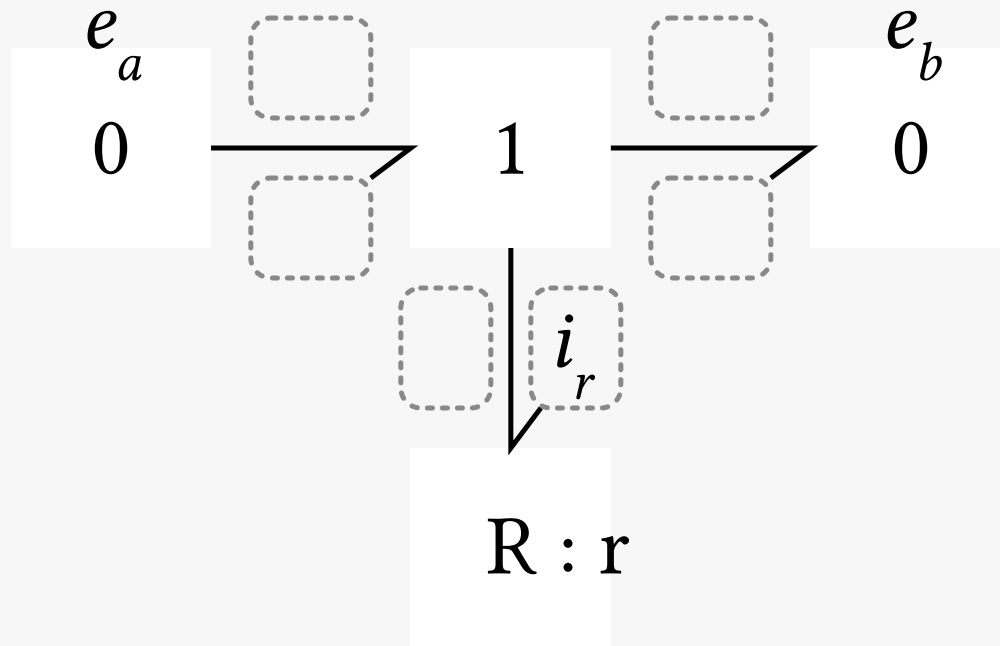
Questions:

- Fill the blanks on the BG
- Represent the physical system

E1: 0 junctions represent voltage nodes (Sol.)



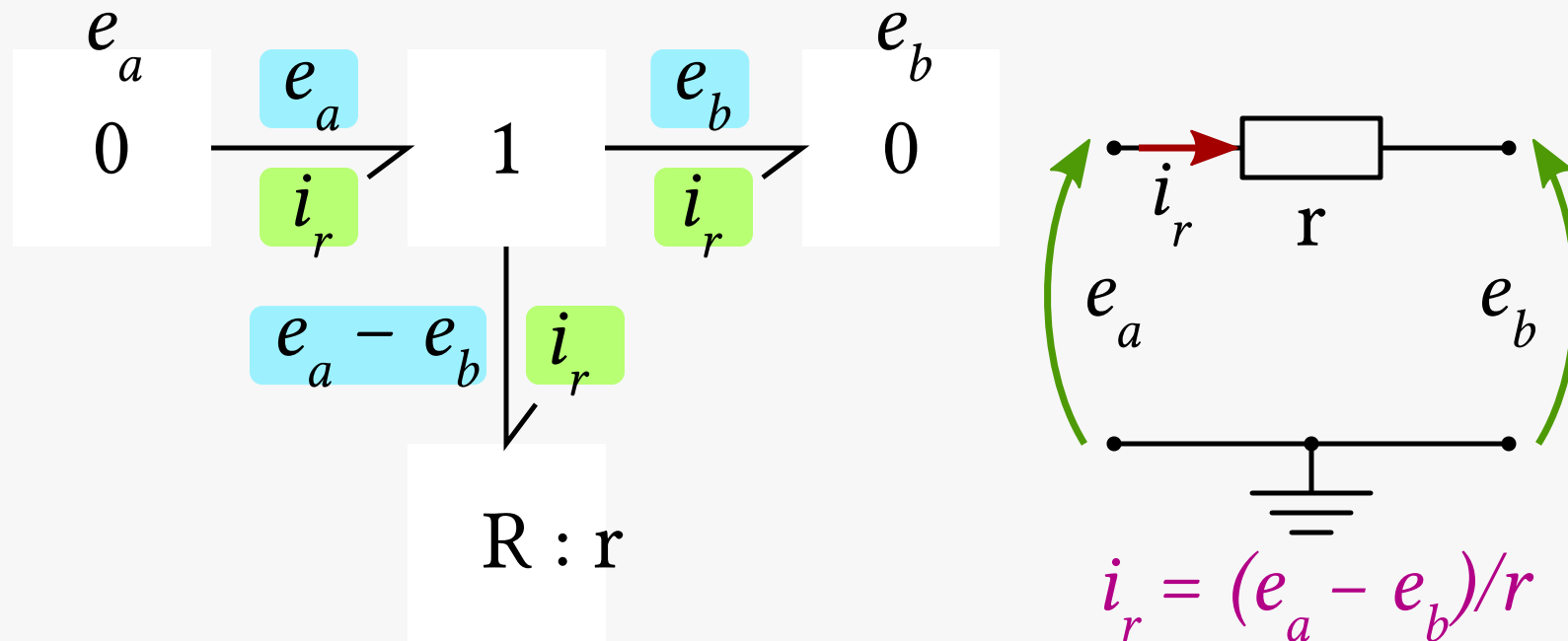
E2: 1 junction for voltage drop on a RLC component



Questions:

- Fill the blanks on the BG
- Represent the physical system
- Compute i_r

E2: 1 junction for voltage drop on a RLC component (Sol.)



$$i_r = (e_a - e_b)/r$$

E3: Voltage source

...

Bond Graphs practice

A graphical language for the analysis of multiphysical systems

→

Pierre Haessig, CentraleSupélec (campus of Rennes)

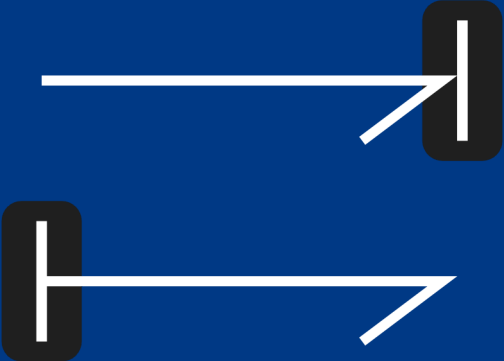
SG6: Dec. 2021 – Jan. 2022

Slide deck 3: causality

Course outline

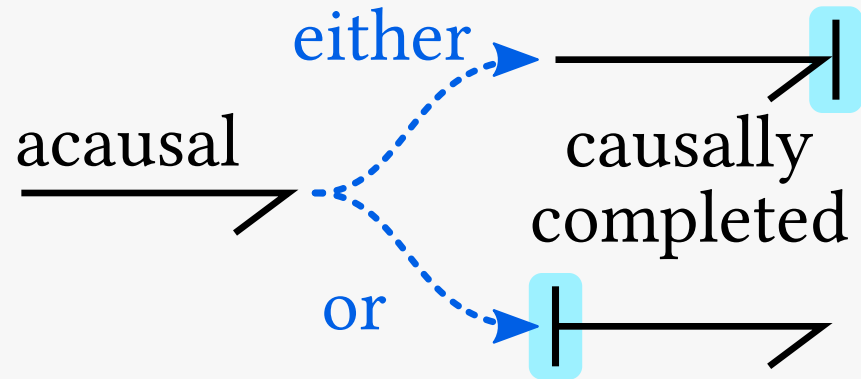
- Bond graph objectives
- The bond graph language
 - Bonds and power variables: the physical analogy
 - Elements
- Practice: reading & creating bond graphs
- Causality and derivation of mathematical models
 - **Principles**
 - Practice

Computational causality & derivation of mathematical models



Causally completed bonds

Reminder: bonds express acausal physical links between components

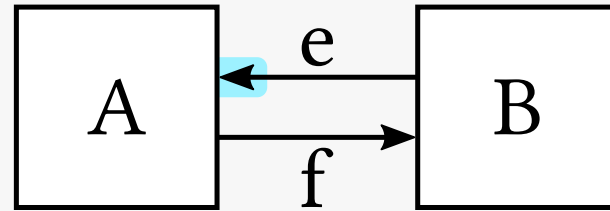
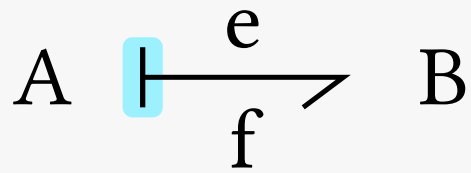
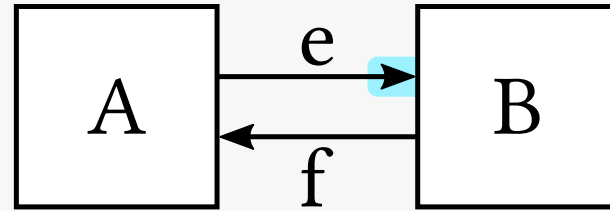
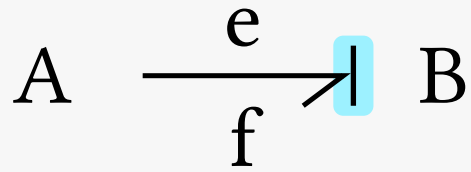


Optionnaly, bonds can be completed with “**causality stroke/arrow**” which indicate the direction of the computation flow.

Remark: this *computational* causality doesn't express a *physical* cause → consequence relationship

Meaning and significance of causality in BGs

BG to block diagrams fragments



Mnemonic: the causality arrow is the direction of the *effort signal arrow*

To be continued...

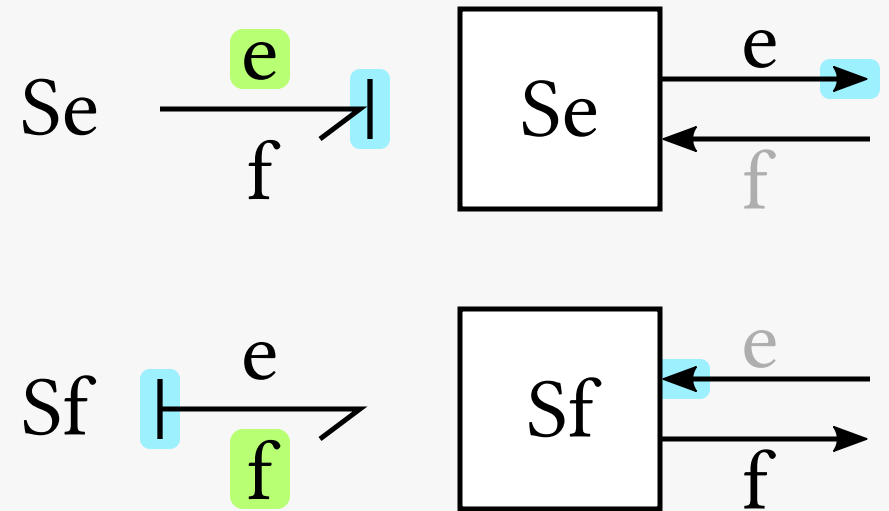
Rules for each class of BG components

- sources
- junctions
- energy stores & dissipators

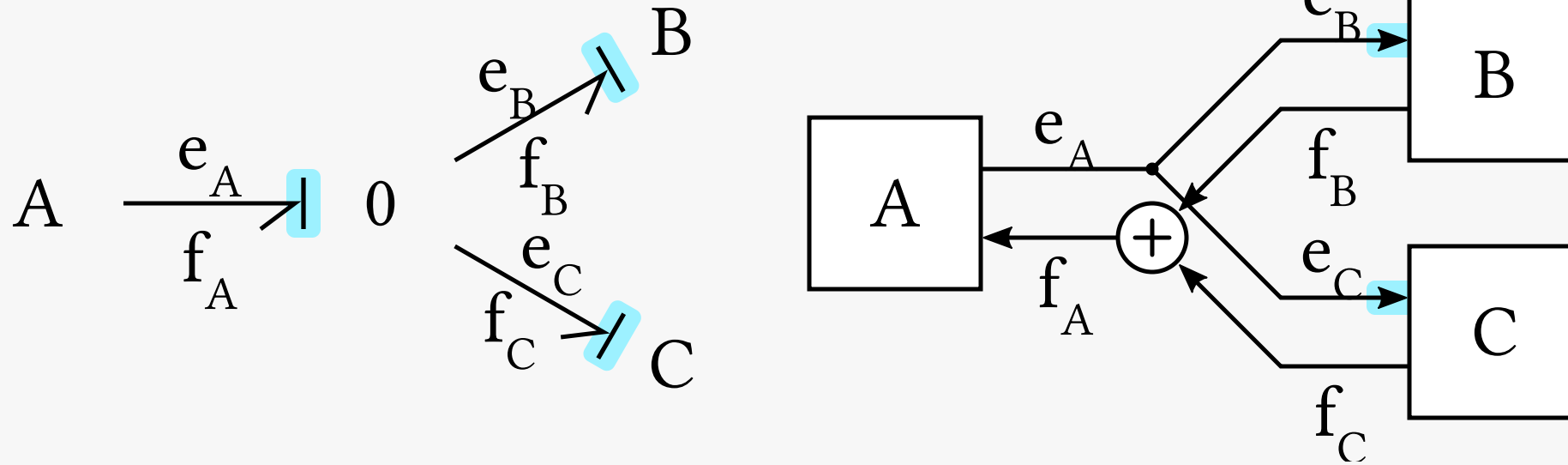
Causality for Sources

For sources, causality orientation is **compulsory**:

- S_e imposes the effort
→ outgoing effort
- S_f imposes the flow
→ outgoing flow



Causality for junctions: 0 junction

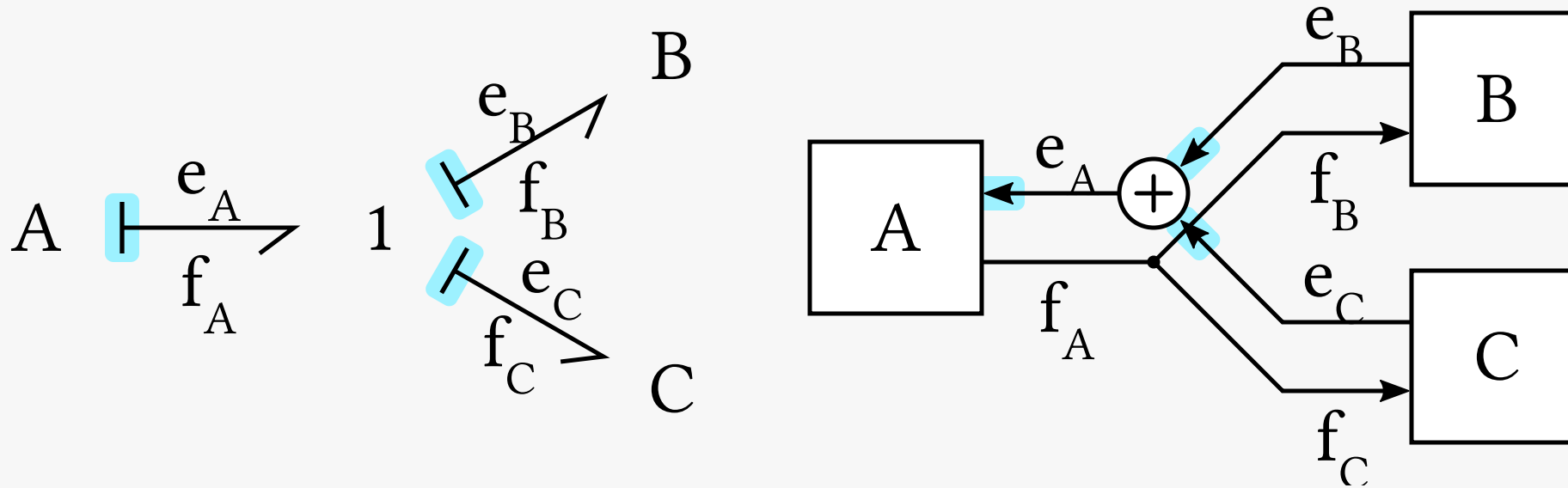


Compulsory rule

For 0 junction (share effort): *“One and only one incoming effort”*

Remark: causality arrows unrelated to bond orientations

Causality for junctions: 1 junction

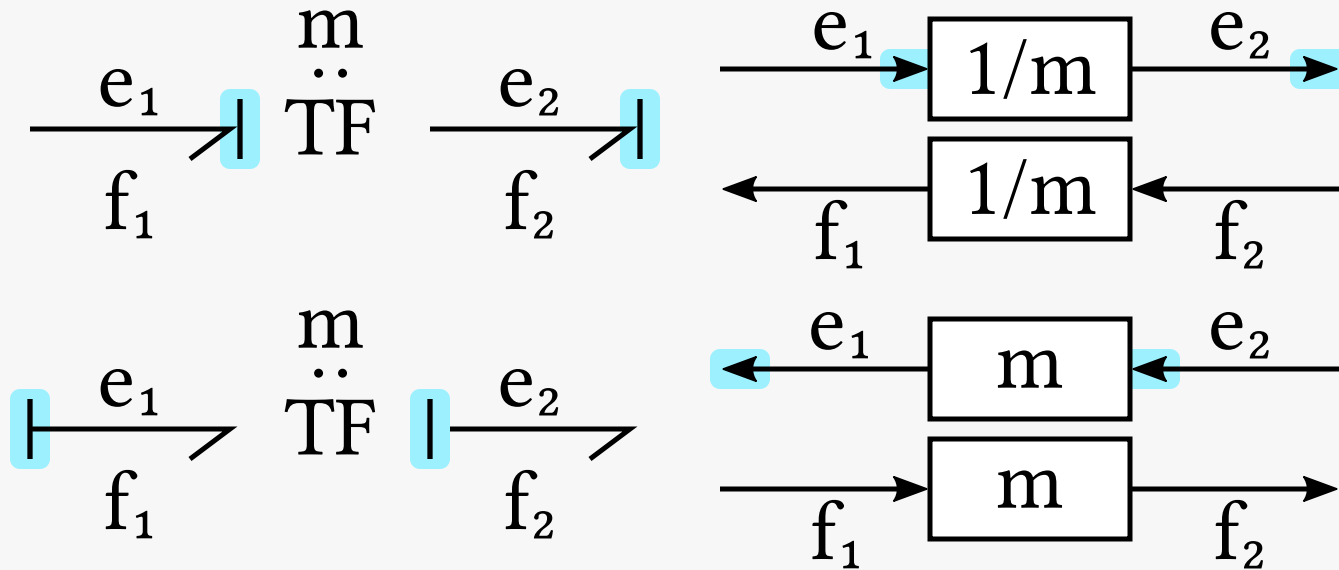


Compulsory rule

For 1 junction (share flow): *“One and only one incoming flow”*

Remark: causality arrows unrelated to bond orientations

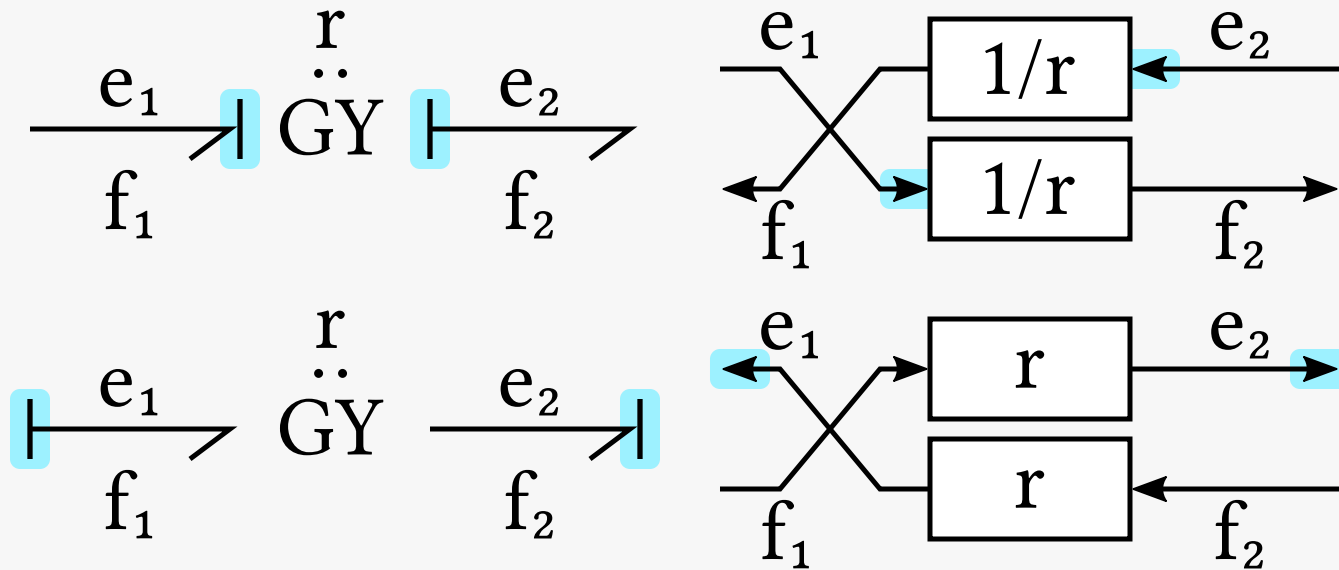
Causality for junctions: TF



[Compulsory] Transformers **preserve** the direction of the e&f signals

- one incoming effort
- one incoming flow

Causality for junctions: GY



[Compulsory] Gytrators **reverse** the direction of the e&f signals

- either two incoming efforts
- or two incoming flows

Causality for Energy stores

C and I energy stores can accept the two possible causalities:

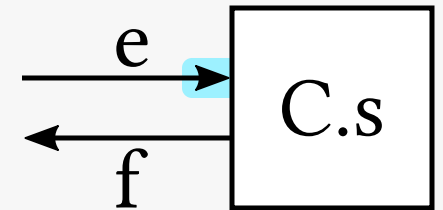
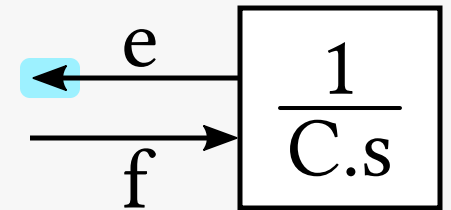
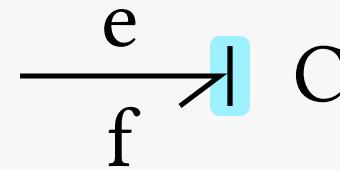
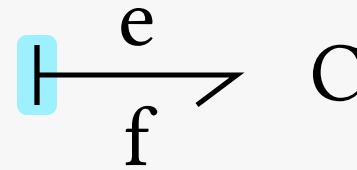
- **Integral causality** is the one yielding an integrator block
- **Derivative causality** is the one yielding an derivator block

For computational easiness, integrators are preferred, so
Integral causality = **Preferred** causality

Causality for Energy stores: C

Integral causality: receives a flow, imposes an effort

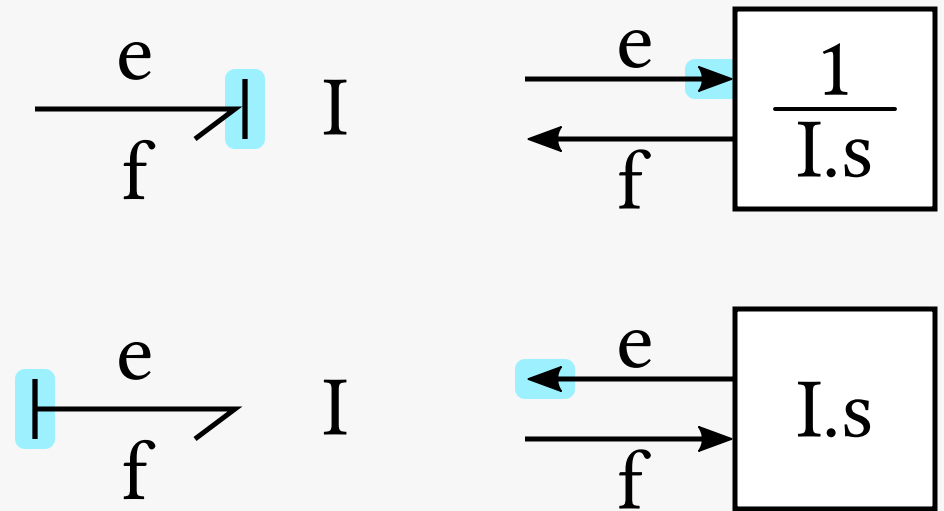
- Capacitor: current \rightarrow voltage
- Spring: speed/position \rightarrow force



Causality for Energy stores: I

Integral causality: receives a flow, imposes an effort

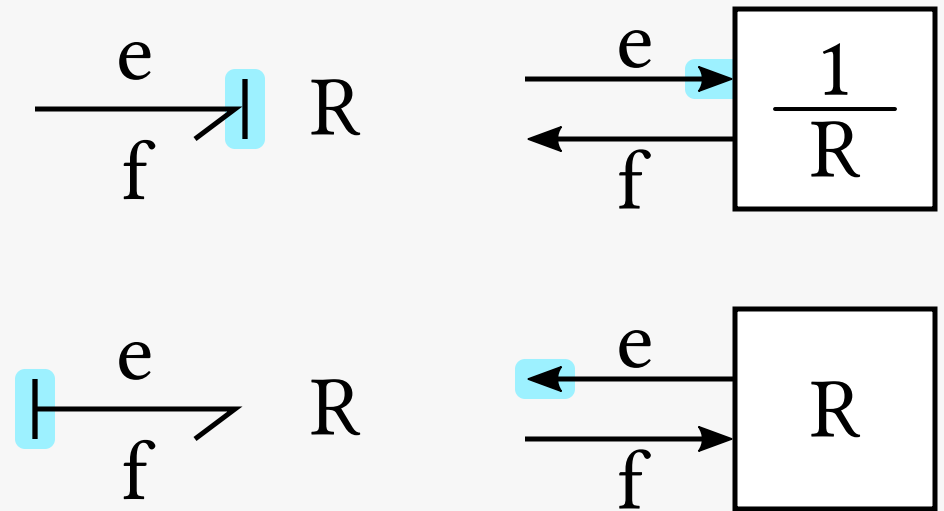
- Inductor: voltage \rightarrow current
- Intertia: force \rightarrow speed



Causality for R (dissipators)

For dissipators, all orientations are fine.

Sometimes called “resistance” vs
“conductance” causality



Sequential assignement rules

SCAP procedure:

Rule 1: assign sources

Rule 2: assign integral causality to Energy stores

Causality assignment examples

- Ex1: Rule 1 (sources+propagation) is enough
- Ex2: Rule 1+2 (preferred causality for energy stores) are enough

Bond Graphs practice

A graphical language for the analysis of multiphysical systems



Pierre Haessig, CentraleSupélec (campus of Rennes)

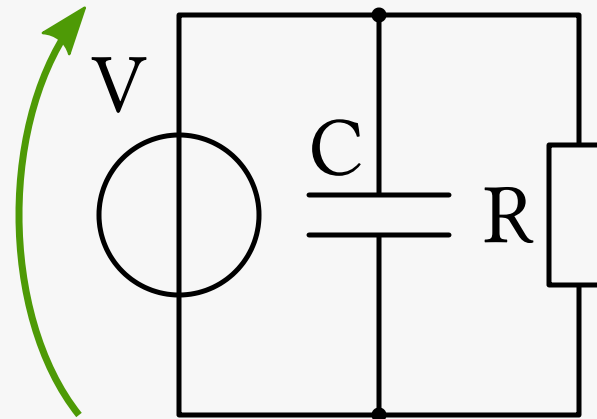
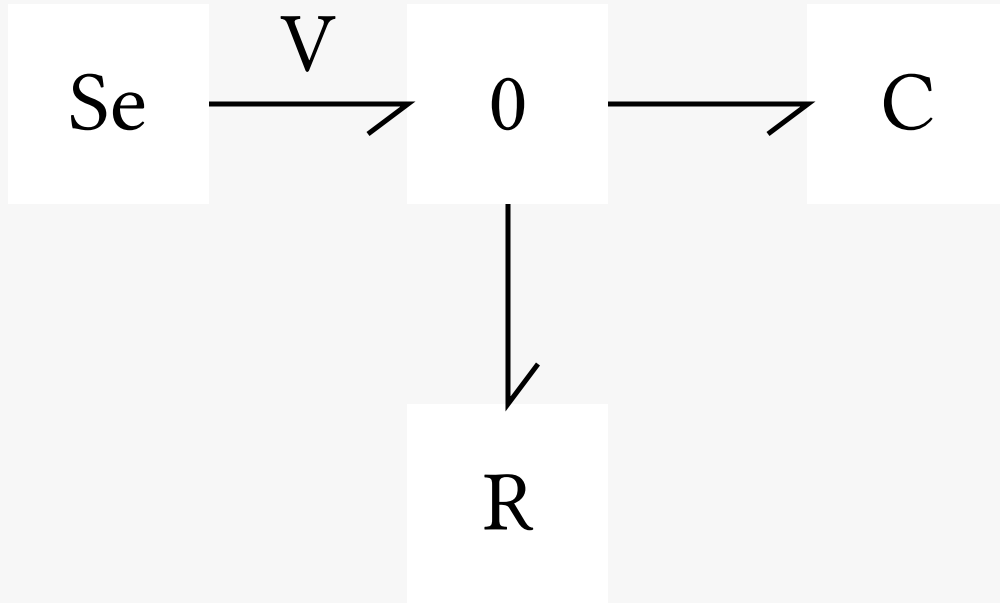
SG6: Dec. 2021 – Jan. 2022

Slide deck 4: causality practice

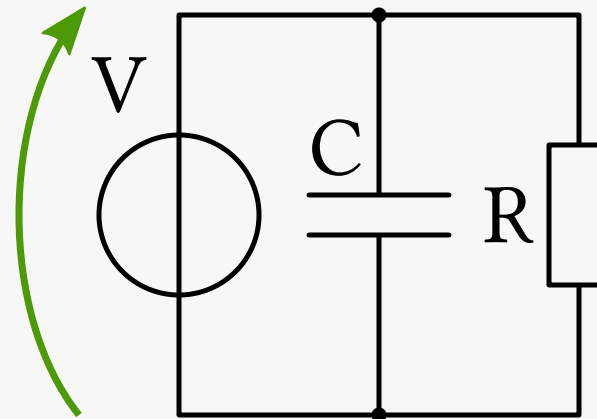
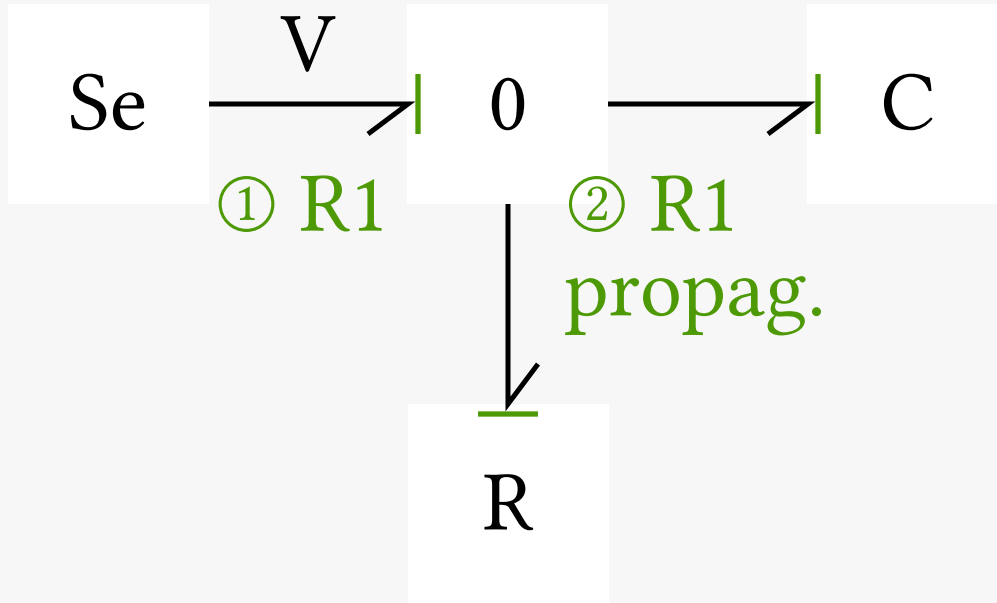
Course outline

- Bond graph objectives
- The bond graph language
 - Bonds and power variables: the physical analogy
 - Elements
- Practice: reading & creating bond graphs
- Causality and derivation of mathematical models
 - Principles
 - **Practice**

Causality assignment Ex1

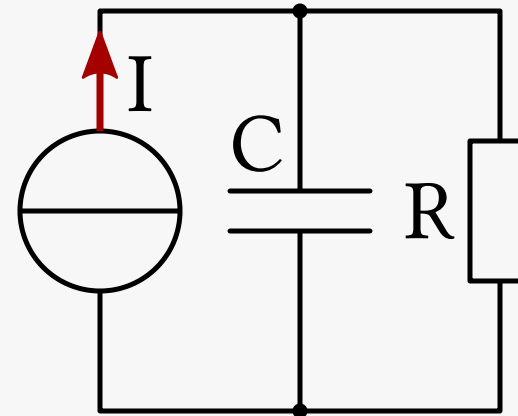
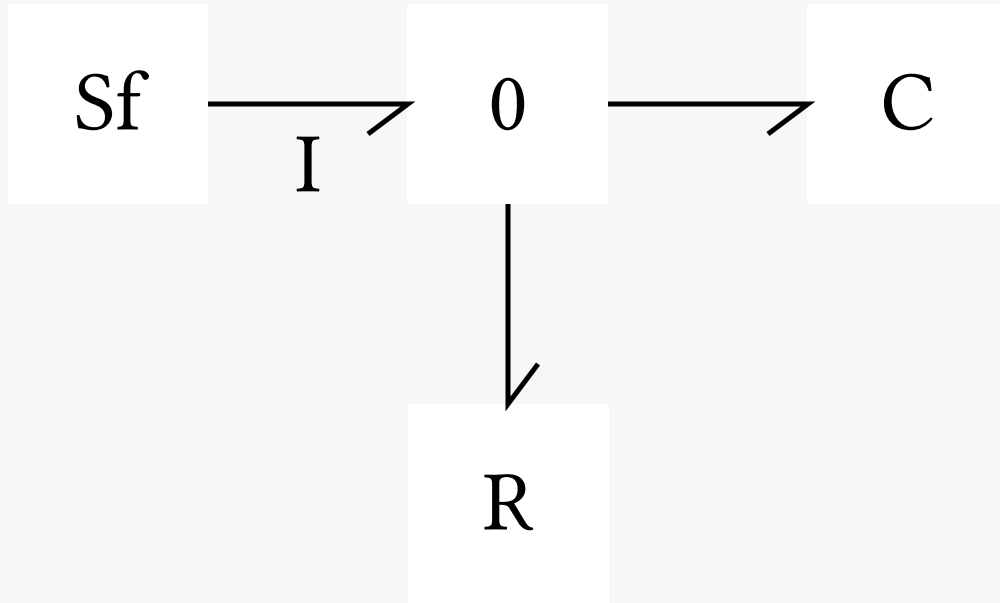


Causality assignment Ex1 (Sol.)

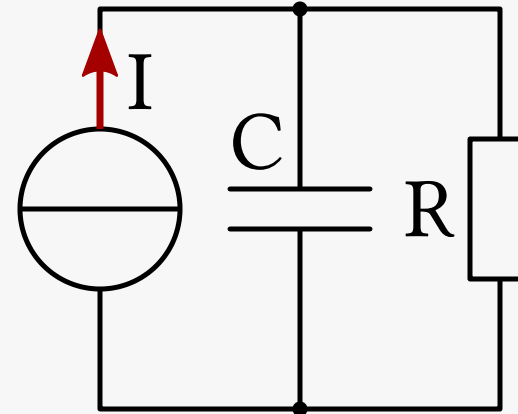
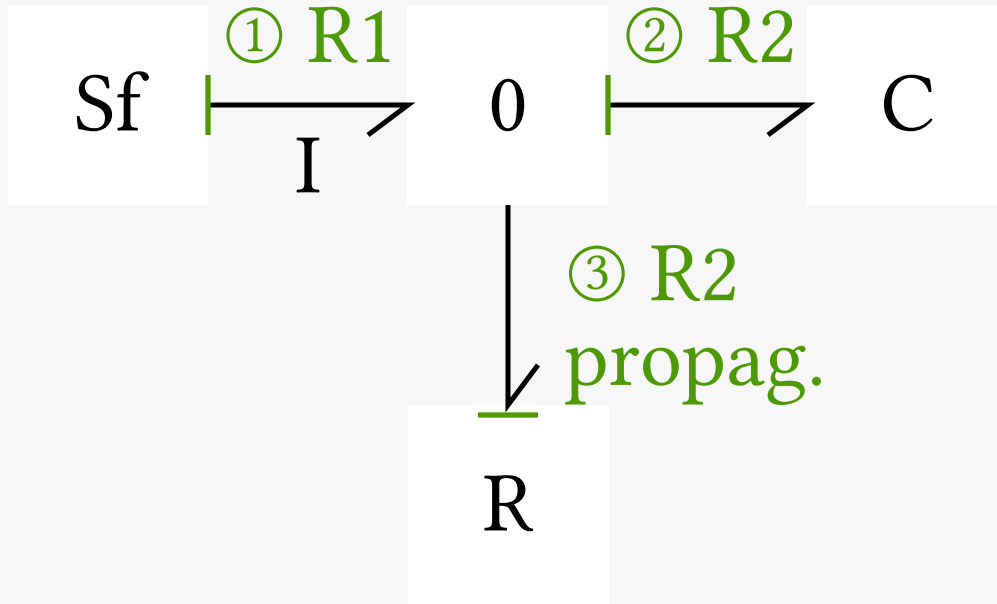


1. R1: Se sets the effort on the 0 junction
2. Propagation: effort propagates through the 0 junction

Causality assignment Ex2



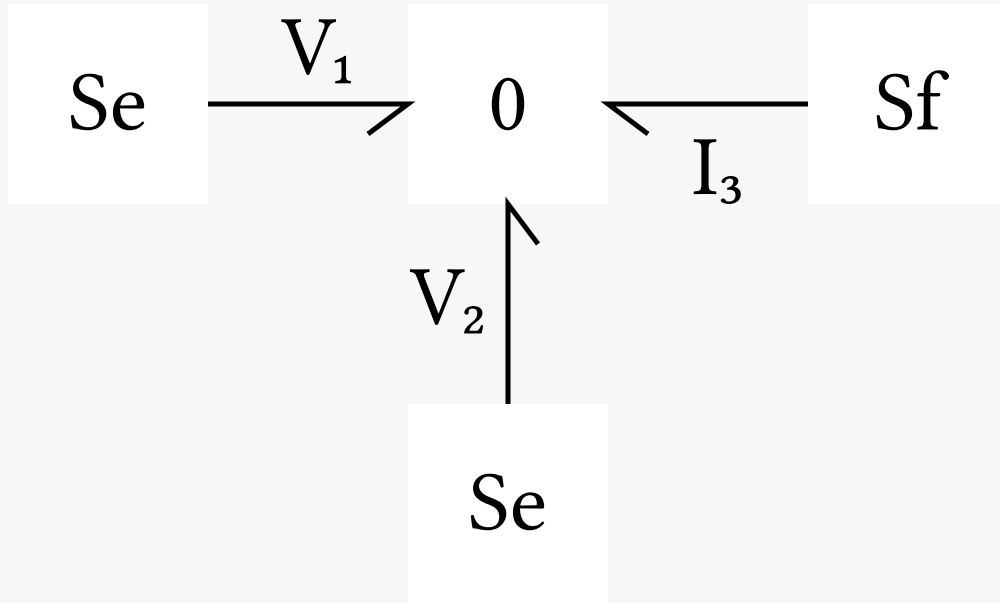
Causality assignment Ex2 (Sol.)



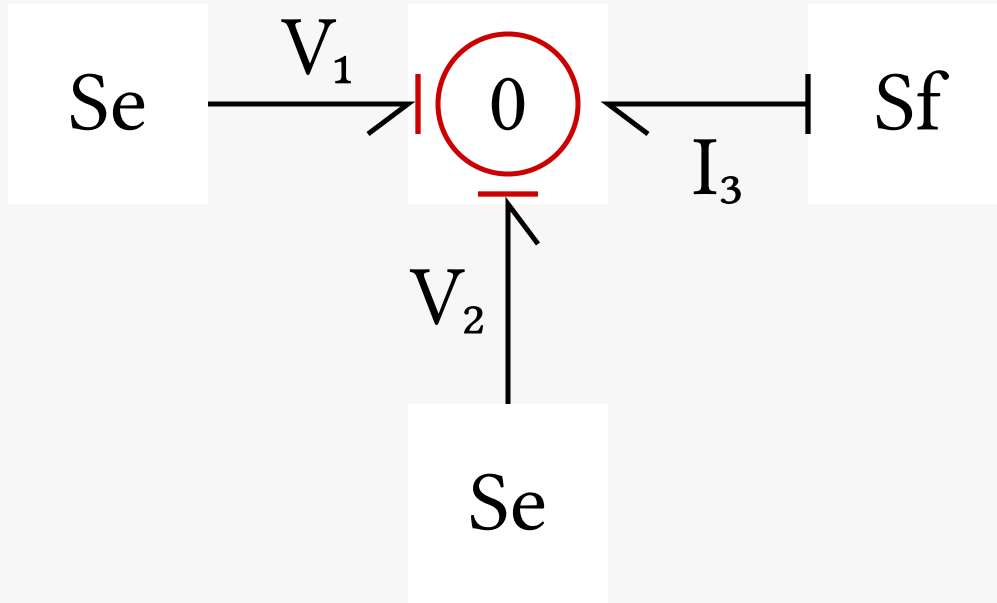
1. R1: Sf sets the flow on the 0 junction.
 - (no propagation)
2. R2: Set integral causality for C
 - \rightarrow C sets the effort on the 0 junction

3. Propagation of effort to R through the 0 junction

Impossible connections exercise



Impossible connections exercise (Sol.)



The physical impossibility of parallel voltage sources emerges as a **causality conflict** on the 0 junction (“only one incoming effort allowed”).